



RECEIVED

JUL 21 2004

Technology Center 2100

Amendments to the Claims:

This listing of claims will replace all prior versions, and listing of claims in the application:

Listing of the Claims:

1. **(Currently Amended)** A method of sending a message from a first common Object Request Broker to a second common Object Request Broker operating in a distributed object oriented environment, said method comprising:

determining whether the message which is to be sent from a first common Object Request Broker to a second common Object Request Broker should be fragmented into two or more sub-messages;

initiating construction of a sub-message which is to be sent from a first common Object Request Broker to a second common Object Request Broker when said determining determines that said message is to be sent in two or more sub-messages;

initializing an offset-variable for said message to zero when said determining determines that said message is to be fragmented into two or more sub-messages;

determining whether there is a need to know the position of a byte of the sub-message with respect to the message;

reading the offset-variable for said message when said determining determines that there is a need to know the position of a byte of the sub-message with respect to the message;

completing construction of the sub-message based on the offset-variable for said message;

updating the offset-variable for said message; and

sending a constructed sub-message, which has been constructed based on the offset-variable, from the first common Object Request Broker to a second common Object Request Broker.

2. (Original) A method as recited in claim 1,
wherein said sub-message has a header that includes one or more bytes, and
wherein said updating of the offset-variable comprises:
subtracting the length of the header of the sub-message from the length of
another sub-message that was constructed immediately prior to construction of the sub-
message.
3. (Original) A method as recited in claim 2, where said updating further comprises:
updating the offset-variable by adding the result of said subtracting to the value
of the offset-variable;
4. (Original) A method as recited in claim 3,
wherein the message is fragmented into N sub-messages, sub-message 0 to
sub-message N, where N is a positive integer, and
wherein the first sub-message has a header with the same number of bytes as
the header of the message.
5. (Original) A method as recited in claim 4, wherein the updating is performed
based on the summation $(L_{i-1} - H_i)$ taken from $i = 1$ to $i = N-1$, where L_i is the length of
the sub-message i and H_i is the length header of the sub-message i .
6. (Original) A method as recited in claim 5, wherein at least two of the headers of
the sub-messages 1 to N can be of different lengths.
7. (Original) A method as recited in claim 1, wherein the method further comprises:
obtaining a remote object; and
invoking a method associated with the object.

8. (Original) A method as recited in claim 7,

wherein said obtaining of the remote object and said invoking of a method associated with the object is performed by a client operating in the distributed object oriented environment, and

wherein the first Object Request Broker creates a request and marshals in appropriate parameters.

9. (Previously Presented) A computing system operating in a distributed object oriented environment, said computing system comprising:

a first common Object Request Broker operating to send a message to a second common Object Request Broker, said message being fragmented by the first common Object Request Broker into two or more sub-messages in a sequence, [[and]]

a message fragment offset-variable provided for said first common Object Request Broker, wherein the message fragment offset-variable indicates a position of a byte of a sub-message with respect to the message, and wherein said position can be determined based on said message fragment offset-variable by subtracting the length of the header of the sub-message from the length of another sub-message immediately preceding the sub-message, and then adding the result of the subtraction to the value of the message fragment offset-variable.

10. (Previously Presented) A computing system as recited in claim 9, wherein the message is fragmented into N sub-messages, sub-message 1 to sub-message N, where N is a positive integer, and

wherein the first sub-message has a header with the same number of bytes as the header of the message.

11. (Previously Presented) A computing system as recited in claim 10, wherein at least two of the headers of the sub-messages 1 to N can be of different lengths.

12. (Previously Presented) A computing system as recited in claim 10, wherein at least two of the sub-messages have data portions that are of different sizes.

13. (Currently Amended) A method of sending a message from a first common Object Request Broker to a second common Object Request Broker operating in a distributed object oriented environment, said method comprising:

initiating construction of a sub-message that is to be sent from a first common Object Request Broker to a second common Object Request Broker ~~when said determining determines that said message is to be sent in two or more sub-messages;~~

determining whether there is a need to know the position of a byte of the sub-message with respect to the message;

reading the an offset-variable when said determining determines that there is a need to know the position of a byte of the sub-message with respect to the message;

completing construction of the sub-message based on the offset-variable; and

updating the offset-variable by subtracting the length of the header of the sub-message from the length of another sub-message that was constructed immediately prior to construction of the sub-message and adding the result of the subtraction to the value of the offset-variable.

14. (Currently Amended) A method as recited in claim 13, wherein the first sub-message is constructed with ~~has~~ a header that has ~~with~~ the same number of bytes as the header of the message.

15. (Original) A method as recited in claim 13, wherein at least two of the headers of the sub-messages can be of different lengths.

16. (Original) A method as recited in claim 13, wherein at least two of the sub-messages have data portions that are respectively of different sizes.

17. **(Currently Amended)** A computer readable media including computer program code for sending a message from a first common Object Request Broker to a second common Object Request Broker operating in a distributed object oriented environment, said method comprising:

computer program code for determining whether the message which is to be sent from the first common Object Request Broker to the second common Object Request Broker should be fragmented into two or more sub-messages;

computer program code for initiating construction of a sub-message when said determining determines that said message is to be sent in two or more sub-messages;

computer program code for initializing an offset-variable to zero when said determining determines that said message is to be fragmented in two or more sub-messages;

computer program code for determining whether there is a need to know the position of a byte of the sub-message with respect to the message;

computer program code for reading the offset-variable when said determining determines that there is a need to know the position of a byte of the sub-message with respect to the message;

computer program code for completing construction of the sub-message based on the offset-variable;

computer program code for updating the offset-variable; and

computer program code for sending a constructed sub-message from the first common Object Request Broker to a second common Object Request Broker.

18. **(Original)** A computer readable media as recited in claim 17,

wherein said sub-message has a header that includes one or more bytes, and

wherein said computer program code for updating the offset variable comprises:

computer program code for subtracting the length of the header of the sub-message from the length of another sub-message that was constructed immediately prior to construction of the sub-message.

19. (Original) A computer readable media as recited in claim 17, wherein said computer program code for updating further comprises:

computer program code for updating the offset-variable by adding the result of said subtracting to the value of the offset-variable.

20. (Original) A computer readable media as recited in claim 17, wherein at least two of the headers of the sub-messages can be of different lengths.

21. (Original) A computer readable media as recited in claim 17, wherein at least two of the data portions of the sub-messages can be of different lengths.

22. **(Currently Amended)** A method of sending a message from a first common Object Request Broker to a second common Object Request Broker operating in a distributed object oriented environment, said method comprising:

providing a message fragment offset-variable for the first common Object Request Broker, wherein the message fragment offset-variable indicates an offset of the message **that is to be sent by the first common Object Request Broker to a second common Object Request Broker;**

determining whether the message is to be fragmented in two or more sub-messages;

initiating construction of a sub-message when said determining determines that said message is to be sent in two or more sub-messages;

completing construction of the sub-message based on the message fragment offset-variable to generate a constructed sub-message; and

sending the constructed sub-message from the first common Object Request Broker to a second common Object Request Broker.

23. (Previously Presented) A method as recited in claim 22, wherein said method further comprises:

updating the fragment offset-variable by subtracting the length of the header of the sub-message from the length of another sub-message that was constructed immediately prior to construction of the sub-message and adding the result of the subtraction to the value of the offset-variable.